

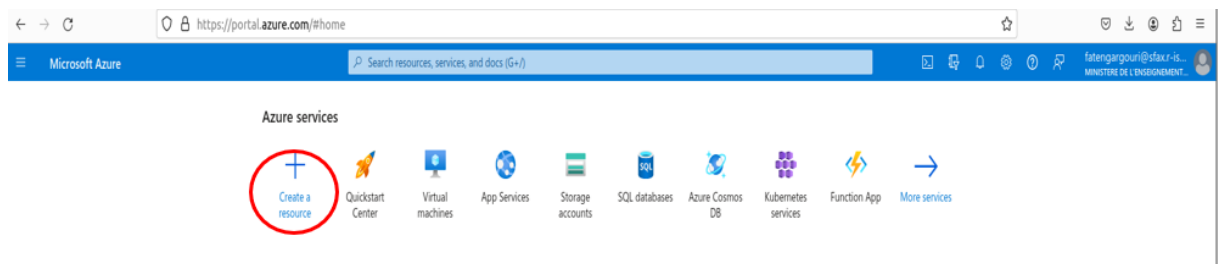
Documentation projet << Déploiement d'une Application sur AKS>>

Ce projet comporte 3 étapes :

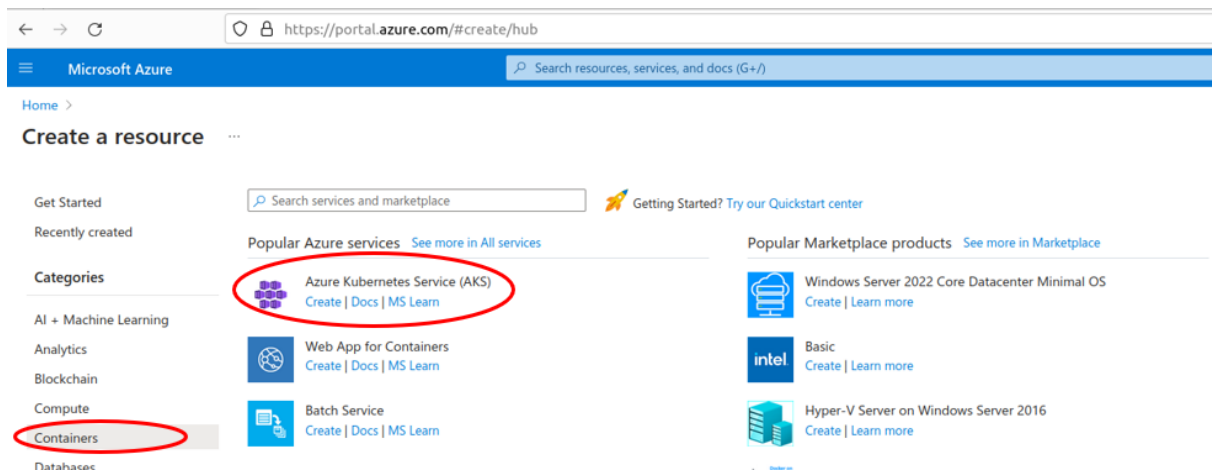
- Configuration et Gestion d'un Cluster AKS dans Azure.
- Publication des images sur Dockerhub
- Déploiement au sein du Cluster Azure Kubernetes Service.

1. Configuration et Gestion d'un Cluster AKS dans Azure.

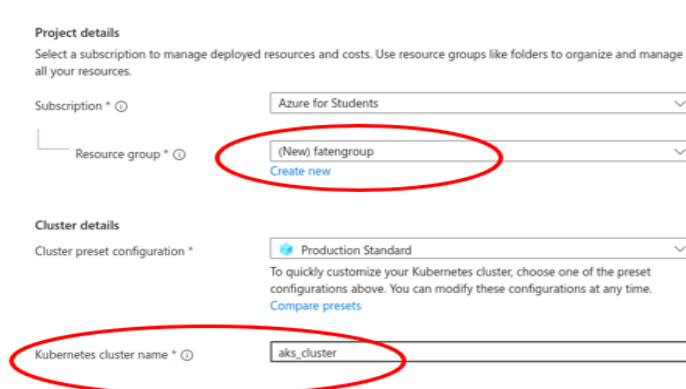
- Se Connecter à Azure
- Cliquer sur Créer une ressource



- Choisir Containers et cliquer sur créer azure kubernetes Service



- Créer un ressource groupe et entrer le nom de kubernetes cluster



- Déterminer la configuration de kubernetes cluster

Kubernetes cluster name *

Region *

Availability zones

AKS pricing tier

Kubernetes version *

Automatic upgrade

Choose between local accounts or Azure AD for authentication and Azure RBAC or Kubernetes RBAC for your authorization needs.

Authentication and Authorization

Once the cluster is deployed, use the Kubernetes CLI to manage RBAC configurations. [Learn more](#)

[Previous](#) [Next](#) [Review + create](#)

Basics Node pools **Networking** Integrations Monitoring Advanced Tags Review + create

Public access

Set authorized IP ranges ☐

Container networking

Network configuration ☒ **Kubenet**
Best for smaller node pools. Each pod is assigned a logically different IP address from the subnet for simpler setup

☐ **Azure CNI**
Best for larger node pools. Each node and pod is assigned a unique IP for advanced configurations

Bring your own virtual network ☐

DNS name prefix *

Network policy * ☒ **None**
Allow all ingress and egress traffic to the pods

☐ **Calico**
Open-source networking solution. Best for large-scale deployments with network security requirements

[Previous](#) [Next](#) [Review + create](#)

- La configuration de cluster est terminé

← → <https://portal.azure.com/#view/HubsExtension/DeploymentDetailsBlade/~/overview/id/%2Fsubscriptions%2Fdc99049d-d8c9-4512-835b-bbf14886efe8%2FresourceGroups%2F...>

Microsoft Azure Search resources, services, and docs (G+)

Home > **microsoft.aks-1704993378001** | Overview

Deployment

Search Delete Cancel Redeploy Download Refresh

Overview

Inputs Outputs Template

✓ Your deployment is complete

Deployment name: microsoftaks-1704993378001 Start time: 1/11/2024, 6:18:24 PM
Subscription: Azure for Students Correlation ID: bfe40eb1-37e2-45f5-9799-8b78e0234d37
Resource group: fatengroup

Deployment details

Next steps

[Go to resource](#)

Cost Management
Get notified to stay within your budget and prevent unexpected charges on your bill.
[Set up cost alerts >](#)

Microsoft Azure Search resources, services, and docs (G+/I)

Home > microsoft.aks-1704993378001 | Overview >

aks_cluster Kubernetes service

Search << + Create Connect Start Stop Delete Refresh Open in mobile Give feedback

Overview

Activity log

Access control (IAM)

Tags

Diagnose and solve problems

Microsoft Defender for Cloud

Kubernetes resources

Namespaces

Workloads

Services and ingresses

Storage

Configuration

Custom resources

Events

Run command

Essentials

Resource group : fatengroup

Status : Succeeded (Running)

Location : East US

Subscription : Azure for Students

Subscription ID : dc99049d-d8c9-4512-835b-bbf14886efe8

Tags (edit) : Add tags

Kubernetes version : 1.27.7

API server address : aks-dns-vd80jgrp.hcp.eastus.azmk8s.io

Network type (plugin) : Kubenet

Node pools : 1 node pool

JSON View

Get started Properties Monitoring Capabilities (4) Recommendations Tutorials

Kubernetes services

Encryption type : Encryption at-rest with a platform-managed key

Virtual node pools : Not enabled

Node pools

Node pools : 1 node pool

Kubernetes versions : 1.27.7

Node sizes : Standard_DS2_v2

Networking

API server address : aks-dns-vd80jgrp.hcp.eastus.azmk8s.io

Network type (plugin) : Kubenet

Pod CIDR : 10.244.0.0/16

Service CIDR : 10.0.0.0/16

DNS service IP : 10.0.0.10

Docker bridge CIDR : -

Network Policy : Calico

2. Publication des images sur Dockerhub :

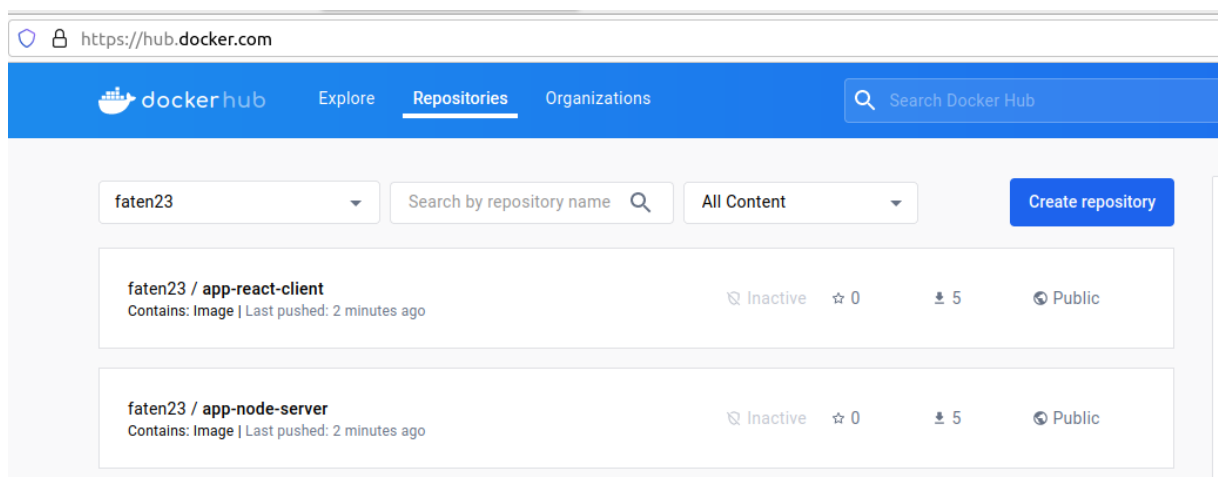
- Créer des fichiers dockerfile et docker compose
- Créer des images docker

```
faten@faten-virtual-machine:~/Bureau/project_cloud/Dockerisation_application$ docker-compose build
mongo uses an image, skipping
Building server
[+] Building 3.3s (12/12) FINISHED
=> [internal] load .dockerignore                                0.1s
=> => transferring context: 2B                                  0.0s
=> [internal] load build definition from Dockerfile             0.1s
=> => transferring dockerfile: 321B                             0.0s
=> [internal] load metadata for docker.io/library/node:10.16-alpine 2.4s
=> [auth] library/node:pull token for registry-1.docker.io     0.0s
=> [1/6] FROM docker.io/library/node:10.16-alpine@sha256:77c898d0da5e7bf 0.0s
=> [internal] load build context                                0.0s
=> => transferring context: 396B                                  0.0s
=> CACHED [2/6] RUN mkdir -p /usr/src/app                       0.0s
=> CACHED [3/6] WORKDIR /usr/src/app                           0.0s
=> CACHED [4/6] COPY package*.json ./                          0.0s
=> CACHED [5/6] RUN npm install --silent                       0.0s
=> [6/6] COPY . .                                              0.2s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.1s
=> => writing image sha256:ef40e23dc95b00b07776d9d297cf7a58ab653df9ea4ce 0.0s
=> => naming to docker.io/library/app-node-server              0.0s
Building client
[+] Building 1.3s (10/10) FINISHED
=> [internal] load .dockerignore                                0.0s
=> => transferring context: 2B                                  0.0s
=> [internal] load build definition from Dockerfile             0.0s
=> => transferring dockerfile: 318B                             0.0s
=> [internal] load metadata for docker.io/library/node:10.16-alpine 0.5s
=> [1/5] FROM docker.io/library/node:10.16-alpine@sha256:77c898d0da5e7bf 0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 1.86kB                               0.0s
=> CACHED [2/5] WORKDIR /usr/src/app                           0.0s
=> CACHED [3/5] COPY package*.json ./                          0.0s
=> CACHED [4/5] RUN npm install --silent                       0.0s
=> [5/5] COPY . .                                              0.4s
=> exporting to image                                           0.2s
=> => exporting layers                                           0.1s
=> => writing image sha256:51ad6aef3c47088f09d0f9501ad4dd899feca1f06b8a1 0.0s
=> => naming to docker.io/library/app-react-client             0.0s
```

- Publier des images sur dockerhub

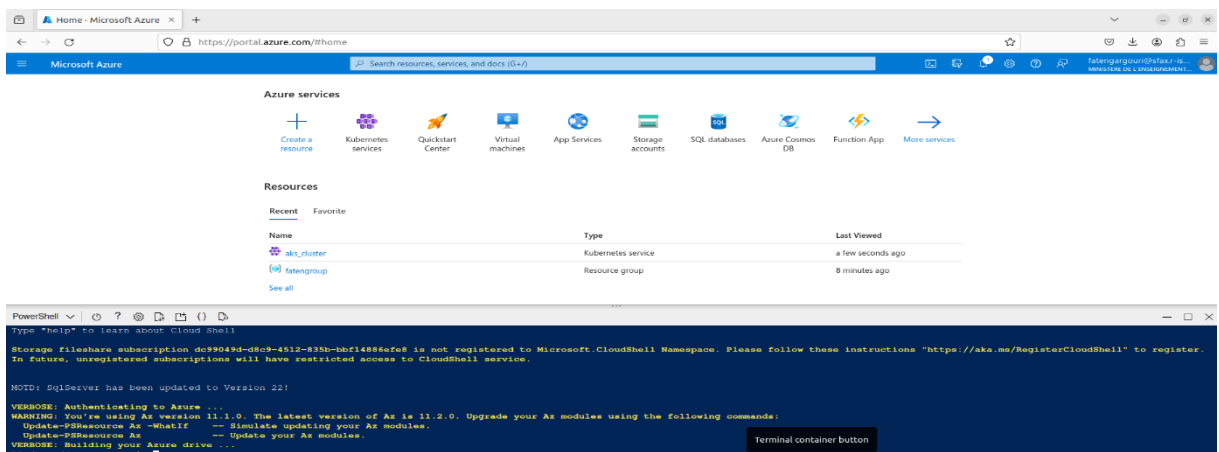
```
faten@faten-virtual-machine:~/Bureau/project_cloud/Dockerisation_application$ docker push faten23/app-node-server
Using default tag: latest
The push refers to repository [docker.io/faten23/app-node-server]
31f1ce9a1949: Layer already exists
858bf0335c4c: Layer already exists
24d280a2f3f1: Layer already exists
5f70bf18a086: Layer already exists
19a7740cafa1: Layer already exists
dd108d5734de: Layer already exists
77306e58a4bd: Layer already exists
a3b85ad42b98: Layer already exists
f1b5933fe4b5: Layer already exists
latest: digest: sha256:41a8f5b114669bf594f1705cef73d733f2868d86c5228894248213d026917d8b size: 2200
faten@faten-virtual-machine:~/Bureau/project_cloud/Dockerisation_application$ docker push faten23/app-react-client
Using default tag: latest
The push refers to repository [docker.io/faten23/app-react-client]
65585fa0138d: Layer already exists
cd14cccc1f11: Layer already exists
c44ae454e22e: Layer already exists
1984fbc1bbe2: Layer already exists
dd108d5734de: Layer already exists
77306e58a4bd: Layer already exists
a3b85ad42b98: Layer already exists
f1b5933fe4b5: Layer already exists
latest: digest: sha256:0e0af4b9819e305c7066055214b4d3e54bdc4ad6dca4efd7b5231d8066797797 size: 1997
```

- Les images sont publiées dans dockerhub



3. Déploiement des images Docker au sein du Cluster Azure Kubernetes Service.

- Ouvrir l'azure cli



- Créer un fichier deployment.yml

```
PS /home/fatengargouri> vi deployment.yml
```

- Ecrire le contenu de fichier deployment.yml
- Exécuter le fichier deployment.yml : `kubectl apply -f deployment.yml`
- Afficher les services :

```
PS /home/fatengargouri> kubectl get services
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
app-node-server-service	LoadBalancer	10.0.117.54	20.253.75.228	80:30389/TCP	40m
app-react-client-service	LoadBalancer	10.0.115.5	20.231.112.80	80:32185/TCP	36m
kubernetes	ClusterIP	10.0.0.1	<none>	443/TCP	143m

- Accéder au site via le navigateur :

